REMARKS

Previously presented claims 1, 13, 25, and 37-53 are all the claims pending in the application.

Claims 1, 13, 25, and 37-53 stand rejected under 35 U.S.C. §102(e) as anticipated by U.S. Patent Application Publication No. 2003/0159112 to Fry.

Applicants respectfully traverse the rejection based on the following discussion.

I. The 35 U.S.C. 102(e) Rejection as Anticipated by Fry

A. The Fry Disclosure

Fry discloses that "[b]road XML support is obtained through use of a set of streaming parser APIs. An application or client needing access to an XML document can contact an XML parser, XML processor, or XML reader in order to gain access to the document. The XML parser selects and instantiates a streaming parser API that is appropriate for the XML document and the client or application. Streaming parser APIs include raw streaming parser APIs, non-validating streaming parser APIs, and validating streaming parser APIs. The XML parser can then provide a variety of types of access to the application or client that does not require the entire document to be read into memory, including providing an XML stream, pulling XML information, and skipping unwanted XML from the document." (Abstract, which is cited by the Final Action).

Fry also discloses that "a programmer can ask for the next event [i.e., next SAX event], or pull the next event, rather than handling the event in a callback. This gives the programmer more procedural control over the processing of the XML document. A streaming parser also allows the programmer to stop processing the document, skip ahead to specific sections of the document, and/or get subsections of the document as mini DOM trees." (Paragraph **[0028]**, which is cited by the Final Action).

Fry further discloses that "Fig. 3 illustrates an event stream, with methods being used to manipulate the current position in the stream. The column on the left represents the XML document, the column on the right represents the Java code, and the column in the middle

represents the event stream. In the Figure, the Java method startDocument( ) is shown to correspond to the Processing Instruction 300 of the event stream, which looks to the header of the XML document. The Java method startElement( ) is called and passed with the event "doc", which corresponds to the StartElementEvent:doc event 302 or the <doc> tag in the XML element. At the first element in the body of the XML document, given here as type "one", a startElement( ) method is again called, but with the element property corresponding to StartElementEvent:element 304 event in the event stream. In the XML document, a./element> end tag signifies the end of the element, corresponding to an EndElementEvent 308in the event stream." (Paragraph **[0029]**, which is cited by the Final Action).

Fry yet further discloses that "the parser would then reach element type "two" in the XML document, corresponding to another StartElementEvent:element 310 in the event stream. this would generate a substream in the Java environment to handle the second element type. Values 312, 314, 316 of element type "two" are placed onto the event stream and correspond to the Java substream. Element type "two" ends when another end tag is reached in the document, corresponding to an EndElementEvent 318 in the event stream, with another EndElementEvent 320 corresponding to the end of the document tag </doc>." (Paragraph **[0030]**, which is cited by the Final Action).

Fry yet further discloses that "[i]n a method for utilizing such an event stream, the name of the element to be extracted from the XML document is passed to an iterative method built on top of a base parser, such as a SAX API or DOM parser. An element tag of the XML document is located and the element type read by the base parser, without necessarily reading the sub-elements or text of that element. The elements of the XML document are then stepped through by the base parser in combination with the iterative method until the element to be extracted is located, read, and processed by the base parser. An event is generated, that is related to the element, and placed on an event stream for use by an application such as a Java application." (Paragraph **[0031]**, which is cited by the Final Action).

B. Arguments

Independent claim 1 recites in relevant part,

"A method for query processing ...

scanning, by a processor, all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers;

parsing a matched textual element, if a tag identifier, corresponding to said matched textual element, matches said query; and

skipping an unmatched textual element for parsing, if a tag identifier, corresponding to said unmatched textual element, does not match said query."

Independent claims 13 and 37 recite in relevant part,

"A system for query processing ...

a processor adapted to scan all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers; and

a parser adapted to ... skip an unmatched textual element for parsing, if a tag identifier, corresponding to said unmatched textual element, does not match said query."

Independent claim 25 recites in relevant part,

"A program storage device readable by computer comprising a program of instructions executable by said computer to perform a method for query processing ...

scanning, by a processor, all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers;

parsing a matched textual element, if a tag identifier, corresponding to said matched textual element, matches said query; and

skipping an unmatched textual element for parsing, if a tag identifier, corresponding to said unmatched textual element, does not match said query."

Fry merely discloses a system and method for XML parsing, in which a name of an element type is extracted from an XML document. An element type tag of the XML document is then located and the element type read by the base parser, without necessarily reading the sub-elements or text of that element. The elements of the XML document are then stepped through

by the base parser in combination with the iterative method until the element to be extracted is located, read, and processed by the base parser.

The present invention describes a system and method of query processing, in which tag identifiers are scanned to determine whether a match exists between any of the tag identifiers and the query, parsing a matched textual element only if the tag identifier matches the query, and skipping parsing of the unmatched textual element, when the tag identifier does not match the query.

In contrast, Fry does not disclose, teach or suggest query processing, as does the present invention. Nowhere does Fry disclose, teach or suggest matching a query to a tag identifier of a mark-up language data stream, as does the present invention. Instead, Fry discloses locating and reading an element type tag of an XML document and then parsing the element by the base parser.

In addition, the present invention describes at least the feature of skipping the parsing of an unmatched textual element, if a tag identifier, corresponding to the unmatched textual element, does not match the query.

In contrast Fry discloses "[a]n element tag of the XML document is located and the element type read by the base parser, without necessarily reading the sub-elements or text of that element." (Paragraph **[0031]**). (emphasis added). That is, Fry always parses the extracted, located, and read element type, but he may not read the sub-elements.

Upon not matching the tag identifier, which might include an element type tag, to the query, however, the present invention skips parsing the unmatched textual element. After all, as is known to those in the art, matching is not parsing. Skipping the parsing relates to the present invention because "there is a need for a novel technique that can reduce parsing time in the context of processing XQuery queries over XML documents" (Specification, page 3, lines 1 and 2).

For at least the reasons outlined above, Applicants respectfully submit that Fry does not disclose, teach or suggest the present invention's features of: "A method for query processing ... scanning, by a processor, all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers; parsing a matched textual element, if a tag

identifier, corresponding to said matched textual element, matches said query; and skipping an unmatched textual element for parsing, if a tag identifier, corresponding to said unmatched textual element, does not match said query", as recited in independent claim 1, "A system for query processing ... a processor adapted to scan all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers; and a parser adapted to ... skip an unmatched textual element for parsing, if a tag identifier, corresponding to said unmatched textual element, does not match said query", as recited in independent claims 13 and 37, and "A program storage device readable by computer comprising a program of instructions executable by said computer to perform a method for query processing ... scanning, by a processor, all tag identifiers of said ordered index to determine if there exists a match between a query and any of said tag identifiers; parsing a matched textual element, if a tag identifier, corresponding to said matched textual element, matches said query; and skipping an unmatched textual element for parsing, if a tag identifier, corresponding to said unmatched textual element, does not match said query", as recited in independent claim 25. Accordingly, Fry does not anticipate, nor render obvious, the subject matter of independent claims 1, 13, 25, and 37, and dependent claims 38-53 under 35 U.S.C. §102(e). Withdrawal of the rejection of claims 1, 13, 25, and 37-53 under 35 U.S.C. §102(e) as anticipated by Fry is respectfully solicited.

II. Formal Matters and Conclusion

Claims 1, 13, 25, and 37-53 are pending in the application.

With respect to the rejection of the claims over the prior art, Applicants respectfully argue that previously presented claims 1, 13, 25, and 37-53 are distinguishable over the prior art of record. Accordingly, the Examiner is respectfully requested to reconsider and withdraw the rejection to the claims.

In view of the foregoing, Applicants submit that claims 1, 13, 25, and 37-53, all the claims pending in the application, are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest time possible.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary.

Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0441.

Respectfully submitted,

Dated: March 24, 2008

/Peter A. Balnave/
Peter A. Balnave, Ph.D.
Registration No. 46,199

Gibb & Rahman, LLC
2568-A Riva Road, Suite 304
Annapolis, MD 21401
Voice: (410) 573-5255
Fax: (301) 261-8825
Email: Balnave@Gibb-Rahman.com
Customer Number: 29154